



## **Advanced software package(s) for modelling luminosity performance.**

E. Adli<sup>1,2</sup>, I. Ahmed<sup>3</sup>, H. Burkhardt<sup>1</sup>, P.N. Burrows<sup>4</sup>, B. Dalena<sup>1</sup>, P. Eliasson<sup>1,5</sup>,  
D. Kruecker<sup>6</sup>, A. Latina<sup>7</sup>, L. Neukermans<sup>1</sup>, F. Poirier<sup>6</sup>, D. Schulte<sup>1</sup>, J. Resta Lopez<sup>4</sup>,  
G. Rumolo<sup>1</sup>, R. Tomas<sup>1</sup>, N. Walker<sup>6</sup>

December, 18 2008

### **Abstract**

A central aim of the last task of WP6 is to provide reliable computer models to study the machine performance with a wide variety of static and dynamic imperfections. To this aim three software packages have been developed and improved. A brief description of these software packages is provided together with the EUROTeV report references, where further details can be found.

---

<sup>1</sup> CERN, Geneva, Switzerland

<sup>2</sup> Oslo University, Norway

<sup>3</sup> CERN-NCP

<sup>4</sup> JAI, Oxford University, UK

<sup>5</sup> Uppsala University, Sweden

<sup>6</sup> DESY, Hamburg, Germany

<sup>7</sup> FNAL, Chicago, USA

## 1 Introduction

Preservation of the ultra-small vertical emittance during the transport and acceleration from the damping ring to the interaction point is one of the major challenges. Very tight tolerances are required on the field quality and alignment of the components, which are beyond those achievable with optical survey techniques. Dynamics effects – such as ground motion and vibration – will have significant detrimental effects on the luminosity and may also significantly impact the initial beam-based alignment and tuning procedures: the ILC-TRC specifically acknowledged this last topic as a high ranking R&D item, noting the need of full simulations of the tuning including the dynamic effects and their compensation. To reliably predict the integrated luminosity performance of the linear collider computer simulations are essential. In order to provide the required reliable computer models to study the machine performance with a wide variety of static and dynamic imperfections, four software packages have been developed and improved:

## 2 PLACET[2]

PLACET is a Tcl-Tk based code (an interface to OCTAVE is also provided). It allows to simulate the transport of the beam in a linear machine: from the damping ring to the IP and beyond taking into account imperfections. It runs by means of user-provided tcl scripts which define the lattice and the beam. It can be used to perform alignment, tuning and feedback of a linear machine. PLACET provides two models of the beam: the particle beam and the sliced beam. The beam is always represented as a number of bunches, each of which is cut longitudinally into slices. Each slice consists of one or more macro particles that have the same longitudinal position. Their energies can differ to simulate the effect of an initial energy spread. If the lattice is linear the particle can also be described by a matrix (see PLACET documentation at <https://savannah.cern.ch/projects/placet/>).

All linear elements are treated as thick elements, but also thin lens approximation is implemented in 4D and 6D ( for particle beam ) phase space coordinates.

It allows to investigate single and multibunch effects. It allows to simulate normal cavities with relatively low group velocities as well as the special transfer structures specific to Clic. In these the group velocity is a significant fraction of the speed of light (around 50%). It allows to simulate the beam dynamics in the presence of wakefields (short and long range) in the main accelerating or decelerating part of a linac as well as in the collimators (short range). A number of correction schemes allow to test the emittance growth to be expected for given prealignment errors.

In the relevant elements the tracking with Incoherent Synchrotron Radiation emission is also considered. In the SBEND the Coherent Synchrotron Radiation can be taken into account too. Total cross-section and luminosity for the colliding beams are evaluated by means of GUINEA-PIG[5].

In the following the main features of PLACET are explained in detail.

### 2.1 Longitudinal Motion

Longitudinal motion has been implemented in PLACET for the following elements: DRIFT, QUADRUPOLE, DIPOLE, SBEND, CAVITY, BPM and MULTIPOLE. All linear elements

are treated as thick elements according to the second order matrix formalism in the full six-dimensional phase space, but thin-lens formalism is also supported for QUADRUPOLE, MULTIPOLE and SBEND elements. Initially, this code was created as an external module in order to provide more flexibility, as well as to enable the development of a version suitable for clusters of computers, as it will be described in the next subsection. Now this code is fully integrated in the PLACET core. The six-dimensional tracking can be enabled (or disabled) for entire segments of lattice by inserting the commands “Begin6d” and “End6d” in the lattice description. Alternatively, it can also be enabled for individual elements using the boolean flag ‘-six\_dim’ at the element declaration. This code is fully operational and it has been used for simulations of both the CLIC and the ILC Bunch Compressors.

Two versions of the Longitudinal Motion tracking module were created: one, just described, was developed for single CPU machines, the other, described here, was designed and optimized for clusters of computers. The implementation of this module took advantage of the Message Passing Interface (MPI) library, a standard protocol for inter-process communications which has been designed for high performance computation on massively parallel machines and on workstation clusters. Parallelizing the code written for the single CPU version was straightforward because the code treats the beam like an array of independent particles (collective effects are not taken into account yet). First the particles are equally distributed between the CPU’s of the cluster, then tracking is performed by each machine and finally the results are gathered together and saved. In order to give an idea of the increment in speed, one should consider that a tracking executed on a cluster composed of N CPU’s will benefit of a speed up by a factor of N. Similarly to the single processor version, the user can enable and disable this module by inserting the commands “MPI Begin6d” and “MPI End6d” at any point in the lattice description.

## 2.2 Emittance/Luminosity Tuning Bumps

A module for simulation of tuning bumps has been developed. The user must define a number of knobs and a tuning signal, before the optimization can start. Knobs can modify quadrupole position, dipole strength, etc. and the tuning signal, which will be optimized, can be drawn from the emittance (either locally or at the end of the linac), from the luminosity, or from any merit function the user defines. The optimization of the settings is carried out by first taking a number of measurements for different knob settings, then a parabolic fit is performed on the obtained values which gives an estimate of the optimal knob settings. In this way one knob after the other can be optimized. The optimization of all knobs can be iterated.

## 2.3 Orbit Correction Algorithms

A number of different alignment techniques is available in the program: One-To-One correction, Few-To-Few correction, Dispersion Free Steering, Ballistic Alignment. In One-to-One correction each corrector is moved to center the beam in the downstream BPM. In Dispersion Free Steering one or more test beams with different energy than the nominal one are steered to follow the same trajectory in order to remove dispersion from the lattice. At the same time one-to-one correction of the nominal beam is used to straighten the nominal orbit. In Ballistic Alignment the quadrupoles are divided into bins which are corrected one after the other. In the first step, the quadrupoles in a bin are switched off to use a ballistic beam to align the beam position monitors (BPMs). Next, a simple one-to-one correction is used to align the quadrupoles.

All the alignment procedures, in the current stage, usually are in the form of test routines that perform an averaging over a number of machines. While it is of course possible to simulate single machine with these routines it is generally much faster to do a significant number in one call since the overhead to calculate the response coefficients needed for the simulation is in most cases larger than time needed to simulate a machine.

A new module introducing the MICADO orbit correction algorithm has been added to PLACET's existent Linear System Solver (LS) correction algorithm. This module is now operational and has been tested in the feedback loop of a dynamic simulation of a CLIC linac which is misaligned due to the ground motion. The results are in accordance with expectations. The PLACET Orbit Correction package was further extended by introducing an orbit correction module based on the Kalman Filter. The Kalman Filter descends from the digital control theory formalism and consists of a set of mathematical equations that provides an efficient computational tool to estimate the state of a process. The Kalman Filter is very powerful in several aspects: it supports estimations of past, present, and even future states, it takes into account the noise in the measurement vectors and it does so even when the precise nature of the modeled system is unknown. In our implementation, the Kalman Filter is used to support the standard LS algorithm that was already present in the code. More precisely, the vector of measured values (i.e. the state of the machine, in this case the vector with the BPM readings) which was feeding the existing LS algorithm, is now replaced by the Kalman Filter's prediction for it which takes into account both the history of the system and the noise affecting the measurements. The convergence of this improved KF+LS algorithm seems to be faster than the plain LS implementation.

## 2.4 Ground motion

A program to include ground motion in PLACET has been developed. This ground motion package is based on a model provided by Andrei Seryi [7] and calculates the misalignment along a given part of the machine. All the data exchange with PLACET is performed via files. The program needs two files: one containing the information about the amplitudes, frequencies and wavelength of the ground motion modes, and another that specifies the longitudinal locations of the points at which the misalignment must be calculated. Based on these two files, the program outputs the misalignments that PLACET reads to misalign the beamline. A seed for the random number generator can be specified to ensure that calls with different positions along the same machine yield consistent motions. In order to be efficient, the program does not calculate the misalignment at a single time but rather at a number of steps in time. A program interfacing PLACET to the "Linear Collider Alignment and Survey" project (see [7]) has been also developed.

## 2.5 Collimator Wake Fields

A module for simulating single bunch collimator wake fields has been constructed and implemented in PLACET. For this purpose, a new element-type, COLLIMATOR, has been added to the set of PLACET lattice definition commands. In order to use it, the user must provide the parameters describing the material and the geometry of the collimator. Based on these numbers, the module can determine the working regime of the collimator, i.e. inductive, diffractive, or intermediate for the geometric wake, and long- or short-range (DC or AC-conductivity) for the resistive wall wake. When tracking, the wake field kick is calculated and applied to the particle. The code evaluates the kick on each particle of the bunch as a function of its longitudinal position (and transverse, if the collimator is flat and there is a quadrupole component of the wake) and applies it. Using this module, a set of simulations has been run to

study the impact of such wake fields on the CLIC luminosity performances. More precisely, we have tracked electrons and positrons through the main linac and then through the BDS, studying the effects both of beam jitters and of collimator misalignments.

## 2.6 Beam Halo and Tail Generator

In order to help for halo and collimation studies, some new features have been implemented in PLACET. A beam-gas scattering fast generator has been interfaced. It simulates elastic and inelastic collisions with beam-gas atoms. Scattered particles are stacked into a secondary beam structure and tracked through the beamline. A new geometrical description of the element aperture allows to localize hits of secondaries in the elements. These hits are now described as hard collimation - where the particle is lost - or soft collimation where particles interact with material through multiple scattering interactions. A photon tracking module has been also implemented. It allows to estimate synchrotron radiation losses through the beam line.

## 2.7 Conversion Tools

In order to benchmark PLACET against other tracking codes, two conversion tools have been created: 1. “mad2placet” fills the conversion from the MAD lattice description language into PLACET and 2. “xsif2placet” converts into PLACET the Standard Input Format (SIF). Our converter is based on the LIBXSIF library, a standalone library for the parsing of Extended Standard Input Format (XSIF) decks in use at SLAC[10]. Besides PLACET, the SIF was also adopted by programs such as LIAR, MAD, DIMAD and TRANSPORT.

The possibility to save the beamline in the Accelerator Markup Language (AML) file format has been added too. AML is a specification of the XML language designed to be used with the Universal Accelerator Parser (UAP). The UAP is a project of the Cornell University to simplify the data exchange among the most diffused codes (XSIF and MAD deck files are supported, for instance)[8].

## 2.8 Accessibility

A web page has been setup to enhance the accessibility of PLACET.

<https://savannah.cern.ch/projects/placet/>

It is based on the LHC Computing Grid (LCG) software development portal and its interface Savannah. The LCG project offers facilities for development, distribution and maintenance of software projects. Savannah is a web-based software hosting system. It includes source management, tracking handles for bugs, tasks, support, news and documentation. All these features make the interaction between the users and the authors more direct and effective, and also improve the coordination between the developers themselves. Many of the benefits of Savannah relies on the Concurrent Versions System (CVS). CVS is a system which allows to record the whole history of each source files and documents and provides tools for downloading the package from the web.

## 2.9 Benchmarking

A number of code-code comparisons have been performed.

In an effort to compare beam dynamics and create a “benchmark” for Dispersion Free Steering (DFS) a comparison was made between different International Linear Collider (ILC) simulation programs while performing DFS[9].

The comparison between the symplectic tracking code SAD with the codes MAD and PLACET in the context of the CLIC beam delivery system shows reasonable agreement between the codes[8].

A throughout benchmarking of the Coherent Synchrotron Radiation (CSR) model implemented in PLACET against Elegant was performed (with the PSI benchmark chicane). The results showed excellent agreement between PLACET and Elegant for CSR in bends and at high energies (for energies below 100 MeV, the results differs substantially due to the  $v = c$  assumption of PLACET). For CSR in drift spaces PLACET includes a simple exponential decay model only [6].

### 3 MERLIN[4]

<http://www.desy.de/~merlin/>

A model from the main linac to the beam delivery system and the interaction point has been set up as a first step towards a full ILC simulation. The model contains wake fields within the accelerating cavities and will allow for realistic alignment and field errors. For the study of correlated ground motion it is necessary to model both, the electron and the positron side, of the ILC. Both halves are defined by independent lattice files which have to be integrated into a common model. Here, the MERLIN library provides the concept of *support structures* which represents an independent layer of geometrical relations on top of the lattice file geometry. Beyond that, a special kind of *support structures (girders)* can be used to define rigid structures, as for example cryomodules or a common support for the magnets of the final focus system.

Our code includes an interface for several ground motion models: a random walk approach (ATL) and models for correlated ground motion. Another interface is implemented to accommodate different beam steering algorithms. This is currently used to realise a one-to-one steering for the horizontal and vertical planes. Fast feedback systems between the main linac and the beam delivery system and for the colliding beams at the IP are part of the simulation although still in a simplified way. The same holds for the tuning of beam parameters at the IP (see below).

MERLIN provides two different bunch types: The *Particle Bunch* class is a collection of 6D phase space coordinates, whereas in the *SMPbunch* (sliced macro-particle) class the transverse coordinates are replaced by 1st and 2<sup>nd</sup> order moments. SMP tracking is faster but not sufficient for higher-order magnetic fields, or in cases where the detailed bunch distribution is of interest (collimation, beam halo etc.). In our model SMP tracking is favourable for both the electron and positron main linacs. When SMP tracking is used in these areas the total (ML and BDS) simulation time is reduced typically by a factor of 5. Finally, GUINEA-PIG [5] is utilised to calculate the total cross-section and luminosity for the colliding bunches including beam-beam effects.

MAD8 [6] is used to define the ILC lattice. Standard accelerator components are defined by MAD keywords, and so-called markers are used to label begin and end of subsystems, or structures as cryo-modules, non-standard components (e.g. undulator) and so forth. In addition names, i.e. character strings, are used to identify individual elements as magnets,

monitors etc. The simulation code has to cope with the fact that naming conventions may change and new elements appear or disappear. Therefore it is crucial for the long-term usability that already the basic program design takes into account potential changes. Wherever possible, string identifier should only be used in isolated places so that the program can be adapted quickly. For special studies or cross-checks the simulation code should be able to track only parts of the ILC lattice, and it should support an easy change between the two different bunch models. Furthermore, a modular accelerator construction should allow a stepwise development of the full simulation.

The model construction starts with parsing the lattice file and the main tasks in this process can be separated into individual classes: reading the file (syntax), understanding the meaning of keywords (semantic) and constructing the corresponding classes (pragmatic). A translation table defines an arbitrary vocabulary of keywords and markers. In Addition, several methods are implemented to modify the parsing behaviour at run time. The accelerator itself is represented by the *Accelerator* class. The class mainly performs loops over a list of *Sub-Systems* (see below) and a separate list of steering algorithms. In addition it exports a list of *support structures* to the ground motion model. The core functionality is provided by the *TypedSubSystem* template class. Depending on the template type (*SMPBunch* or *ParticleBunch*) the class automatically implements the appropriate bunch constructor and tracker class and provides a bunch handler. In general the bunch handler gets the bunch from the previous sub-system, checks if it belongs to the right bunch type and calls, if necessary, the appropriate converter. For the initial sub-system the bunch handler calls a bunch constructor instead. The bunch is then passed to the tracker that propagates the particles through the sub-system. Within this concept the full model is just a line-up of sub-systems and many of technical details on how to implement a simulation model using the MERLIN library are solved in a general way (Fig. 1).

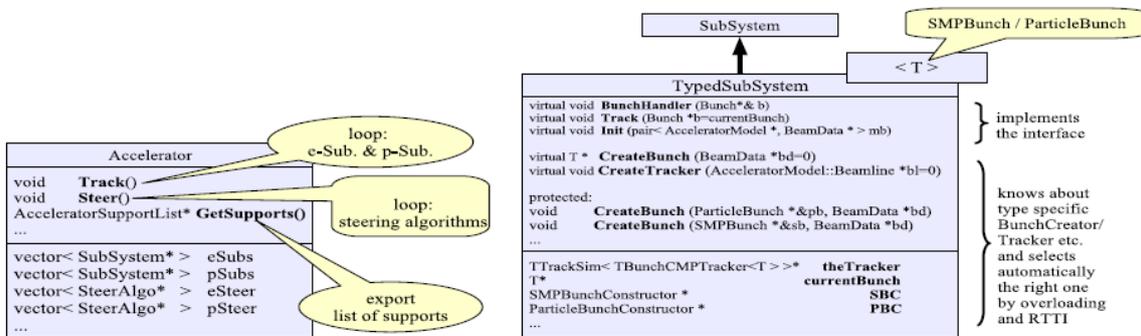


Figure 1: Outline of the *Accelerator* and the *TypedSubSystem* classes.

## 4 FASTION[11]

The fast beam-ion instability is a concern for CLIC. In this process ions from rest gas in the beam pipe are ionized by the beam and then trapped inside the beams. The accumulated ions can lead to a multi-bunch instability. Analytic estimates have been performed and showed that a safe vacuum level is 0.1 nTor in most parts of the machine. In the main linac these estimates suffer from some uncertainty. Due to the required high precision of the components, a safe vacuum level of 0.1 nTor will not be easy to achieve. Such a vacuum quality requires bake-out of the beam line components. The heating of the beam line components is in contradiction with the very tight accuracy requirements arising from beam dynamics. Achieving the 0.1

nTor vacuum level seems very challenging and would need a very ambitious R&D programme. It is therefore important to estimate the required vacuum level more precisely than the analytic estimates allow. A new tracking has been developed to make possible a more quantitative study of the problem. It is specialised on the simulation of the generation of ions and their interaction with the beam. The code represents each bunch of the beam with a large number of macro particles. The production of ions from the rest gas is simulated and the dynamic interaction of these ions with the beams are simulated. The code has been successfully compared to analytic predictions. Its application to the main linac shows that a vacuum level of 10nTor should be sufficient. The code can be found on the savannah server at <https://savannah.cern.ch/projects/fastion> .

## 5 MAPCLASS[12]

The current Final Focus Systems (FFS) of linear colliders requires strong sextupoles to compensate the large chromaticity generated by the strong focusing. This sextupoles introduce higher order aberrations that deteriorate the machine luminosity.

The code MAPCLASS was developed, starting in 2006, to provide a general application able to cancel aberrations of any transfer line based on the computation of the rms beam sizes at the end of the line. This code has been successfully applied to several systems:

- The CLIC FFS [13], where a luminosity increase of 70% was achieved after optimizing the aberrations and the dispersion of the system.
- The CLIC non-linear collimation system [14], where the luminosity was improved by a factor of two by cancelling the residual aberrations of the non-linear elements of this collimation layout.
- The ATF2 nominal design, where the MAPCLASS optimized lattice proved better in tuning simulations. MAPCLASS has been also used to compensate the dodecapolar errors in QF1 by adding a dodecapolar magnet.
- The ATF2 ultra-low beta proposal [15], where a new ATF2 optics with four times smaller vertical beta function was proposed in order to prove CLIC's chromaticity level.

## 6 Conclusion

In the context of the EUROTeV WP6 three advanced software packages have been developed and improved: PLACET, MERLIN and FASTION. Their capability and reliability has been proven through several simulations to study luminosity performance and specific topics of linear machines. A number of studies have been performed for benchmarking against other codes. The codes and their application to the machine performance studies have been documented in several EUROTeV Reports. A summary can be found in EUROTeV Report-2008-090[16]. The codes and documentation have been made publicly available.

## Acknowledgement

This work is supported by the Commission of the European Communities under the 6<sup>th</sup> Framework Programme “Structuring the European Research Area”, contract number RIDS-011899.

## References

- [1] D. Schulte, K. Kubo , ‘Computational Needs for the ILC’, EUROTeV-Report-2006-105.
- [2] A. Latina, H. Burkhardt et al., ‘Recent Improvements of PLACET’, EUROTeV-Report-2006-030.
- [3] A. Latina, D. Schulte et al., ‘Recent Improvements in the tracking code PLACET’, EUROTeV-Report-2008-043.
- [4] D. Krücker, F. Poirier, N.J. Walker, ‘Merlin-based Start-to-End Simulations of Luminosity Stability for the ILC’, EUROTeV-Report-2007-019.
- [5] D. Schulte, Ph.D. thesis, University of Hamburg 1996; TESLA-97-08.
- [6] <http://mad.web.cern.ch/mad/>
- [7] <http://www-pnp.physics.ox.ac.uk/~licas/>
- [8] A. Latina et al. CERN-AB-2008-017, CLIC-Note-759, EUROTeV-Report-2008-043.
- [9] Jeffrey C. Smith et al., SLAC-PUB-13094.
- [10] T. Asakay and J. Resta Lopez CARE-NOTE-2005-006-ELAN, CLIC-Note-637.
- [11] G. Rumolo, D. Schulte, ‘Fast Ion Instability in the CLIC Transfer Line and Main Linac’ EUROTeV-Report-2008-036.
- [12] R. Tomas, ‘MAPCLASS: a code to optimize high order aberrations’ CERN AB-Note-2006-017.
- [13] R. Tomas, ‘Nonlinear optimization of beam lines’ Phys. Rev. ST-AB 9, 081001 (2006); R. Tomas et al., ‘Optimizing the CLIC Beam Delivery System’, EUROTeV-Report-2008-039.
- [14] J. Resta Lopez, ‘Nonlinear Collimation Systems for CLIC and LHC’, PhD Thesis, Universidad de Valencia.
- [15] D. Angal-Kalinin et al, ‘Exploring ultra-low beta\* values in ATF2 - R&D Programme proposal’, CARE/ELAN Document-2008-002
- [16] P.N. Burrows et al., ‘Report on Luminosity Tuning and Control Strategies’ EUROTeV-Report-2008-090.